AIRLINE RESERVATION SYSTEM

REPORT

C++ is an object-oriented programming (OOP) language that is viewed by many as the best language for creating large-scale applications. C++ is a superset of the C language. A related programming language, Java, is based on C++ but optimized for the distribution of program objects in a network such as the Internet.

REPORT OF THE AIRLINE RESERVATION SYSTEM

Airline Reservation System is based on a concept of **booking** and **canceling** flight reservations. Here, the system contains **no login feature**. The users can easily book flights, cancel flight reservation, and check tickets.

MULTIPLE INHERITANCE:

With multiple inheritance, a class is derived from more than one base class. Multiple inheritance is complex and error prone.

this POINTER:

this is a pointer that allows every object to access its own address.

do/while STRUCTURE:

•Similar to while structure.

•Makes loop continuation test at end, not beginning.

•Loop body executes at least once.

- Syntax

```
do {

  statement

  } while ( condition );
```

switch STATEMENT:

A switch statement is a control statement that executes a set of logic based on the result of a comparison between a controlling expression and the labels specified in the switch block.

PROGRAM:

```cpp
#include <iostream>
#include <string>
using namespace std;

// Base class I
class Flight
{
    int flight_number;
    string airline;
    string source;
    string destination;
    int capacity;
    string departure_time;
    string arrival_time;
    int reserved_seats;

    public:
        // Constructor
        Flight(int flight_number, string airline, string source, string destination,
        int capacity, string departure_time, string arrival_time, int reserved_seats)
        {
            this->flight_number = flight_number;
            this->airline = airline;
            this->source = source;
            this->destination = destination;
            this->capacity = capacity;
            this->departure_time = departure_time;
            this->arrival_time = arrival_time;
            this->reserved_seats = reserved_seats;
// here we used "this" data type which means an implicit parameter to all member functions
        }

        // Getter method
        int getFlight_number()
        {
            return flight_number;
        }

        string getAirline()
        {
            return airline;
        }

        string getSource()
        {
            return source;
        }

        string getDestination()
        {
            return destination;
        }
```

```cpp
        int getCapacity()
        {
            return capacity;
        }

        string getDeparture_time()
        {
            return departure_time;
        }

        string getArrival_time()
        {
            return arrival_time;
        }

        int getReserved_seats()
        {
            return reserved_seats;
        }
};

// Base class II
class Passenger
{
    string name;
    int age;
    string gender;
    string contact_info;

    public:
        // Constructor
        Passenger(string name, int age, string gender, string contact_info)
        {
            this->name = name;
            this->age = age;
            this->gender = gender;
            this->contact_info = contact_info;
        }

        // Getter method
        string getName()
        {
            return name;
        }

        int getAge()
        {
            return age;
        }

        string getGender()
        {
            return gender;
        }
```

```cpp
        string getContact_info()
        {
            return contact_info;
        }

        // Function carrying Data
        void data()
        {
            cout << "Enter your Name  : "  << endl;
            cin>> name;
            cout << "Enter your Age : "  << endl;
            cin>> age;
            cout << "Enter your Gender : "  << endl;
            cin>> gender;
            cout << "Enter your Contact INFO : "  << endl;
            cin>> contact_info;
        }

};

// Derived class (multiple inheritance)
class Reservation: public Passenger, public Flight
{
    int reservation_ID;

    public:
        // Constructor
        Reservation(int reservation_ID, string name, int age, string gender, string contact_info, int flight_number,
        string airline, string source, string destination, int capacity, string departure_time, string arrival_time, int reserved_seats) :

            Passenger(name, age, gender, contact_info),

            Flight(flight_number, airline, source, destination, capacity, departure_time, arrival_time, reserved_seats)

            {
                this->reservation_ID = reservation_ID;
            }

        // Getter method
        int getReservation_ID()
        {
            return reservation_ID;
        }

        // Function to book a flight
        void BookFlight()
        {
            cout << "\nAirline : " << getAirline() << endl;
            cout << "Flight No. : " << getFlight_number() << endl;
            data();
            cout << "\nYou have Successfully Booked a Flight!" << endl;
            cout << "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd"
                    "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd" << endl;
        }

        // Function to cancel a reservation
        void CancelReservation()
        {
            cout << "\nEnter Reservation ID : "  << endl;
            cin>> reservation_ID;
            cout << "\nYour Reservation has been canceled successfully!" <<endl;
            cout << "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd"
                    "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd" << endl;
        }

        // Function to display flight
        void displayFlights()
        {
            cout << "\n Flight List \n" << endl;
            cout << "Flight Number: " << getFlight_number() << endl;
            cout << "Source: " << getSource() << endl;
            cout << "Destination: " << getDestination() << endl;
            cout << "Departure Time: " << getDeparture_time() << endl;
            cout << "Arrival Time: " << getArrival_time() << endl;
            cout << "Reserved Seats: " << getReserved_seats() << endl;
            cout << "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd"
                    "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd" << endl;
        }

        // Function to display reservation
        void displayReservations()
        {
```

```cpp
            cout << "\n Reservation List \n" << endl;
            cout << "Reservation ID: " << getReservation_ID() << endl;
            cout << "Flight Number: " << getFlight_number() << endl;
            cout << "Source: " << getSource() << endl;
            cout << "Destination: " << getDestination() << endl;
            cout << "Departure Time: " << getDeparture_time() << endl;
            cout << "Arrival Time: " << getArrival_time() << endl;
            cout << "Passenger Name: " << getName() << endl;
            cout << "Passenger Contact Info: " << getContact_info() << endl;
            cout << "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd"
                "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd" << endl;
    }
};

int main()
{
    cout<<"\n    \xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd"
    "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd"<<endl;
    cout << "\t    AIRLINE RESERVATION SYSTEM \n";
    cout<<"\n    \xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd"
    "\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n"<<endl;
    cout<<"   \xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd MENU OPTIONS \xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n"<<endl;
    Reservation r1(123, "Wareesha Naveed", 19, "F", "03XXXXXXXX", 828, "MONTEGO AIR", "JAMAICA", "NEW YORK", 600, "23:34:01", "02:22:55", 587 );
    int choice;
    do
    {
        cout << "\n1. Book a Flight" << endl;
        cout << "2. Cancel a Reservation" << endl;
        cout << "3. Display Available Flights" << endl;
        cout << "4. Display Reservations" << endl;
        cout << "0. Exit" << endl;
        cout << "\nEnter your choice: ";
        cin >> choice;

        switch (choice)
        {
            case 1:
                r1.BookFlight();
                break;
            case 2:
                r1.CancelReservation();
                break;
            case 3:
                r1.displayFlights();
                break;
            case 4:
                r1.displayReservations();
                break;
            case 0:
                cout << "Exiting the program..." << endl;
                break;
            default:
                cout << "Invalid choice! Please try again." << endl;
                break;
        }
// switch statement is the control statement that allows any value to change the control of the execution.
    }
    while (choice != 0);
// The body of do...while loop is executed once before the condition is checked.
    return 0;
}
```

OUTPUT:

- ### BOOK A FLIGHT:

```
═══════════════════════════════════════
            AIRLINE RESERVATION SYSTEM

═══════════════════════════════════════


═══════════════ MENU OPTIONS ═══════════


1. Book a Flight
2. Cancel a Reservation
3. Display Available Flights
4. Display Reservations
0. Exit

Enter your choice: 1

Airline : MONTEGO AIR
Flight No. : 828
Enter your Name  :
wareesha
Enter your Age :
19
Enter your Gender :
F
Enter your Contact INFO :
03XXXXXXXX

You have Successfully Booked a Flight!
═══════════════════════════════════════
```

- **CANCEL A RESERVATION:**

```
1. Book a Flight
2. Cancel a Reservation
3. Display Available Flights
4. Display Reservations
0. Exit

Enter your choice: 2

Enter Reservation ID :
123

Your Reservation has been canceled successfully!
```

- **DISPLAY AVAILABLE FLIGHTS:**

```
1. Book a Flight
2. Cancel a Reservation
3. Display Available Flights
4. Display Reservations
0. Exit

Enter your choice: 3

 Flight List

Flight Number: 828
Source: JAMAICA
Destination: NEW YORK
Departure Time: 23:34:01
Arrival Time: 02:22:55
Reserved Seats: 587
```

- **DISPLAY RESERVATION:**

```
1. Book a Flight
2. Cancel a Reservation
3. Display Available Flights
4. Display Reservations
0. Exit

Enter your choice: 4

 Reservation List

Reservation ID: 123
Flight Number: 828
Source: JAMAICA
Destination: NEW YORK
Departure Time: 23:34:01
Arrival Time: 02:22:55
Passenger Name: wareesha
Passenger Contact Info: 03XXXXXXXX
```

- **INVALID CHOICE:**

```
1. Book a Flight
2. Cancel a Reservation
3. Display Available Flights
4. Display Reservations
0. Exit

Enter your choice: 5
Invalid choice! Please try again.
```

- **EXIT:**

```
1. Book a Flight
2. Cancel a Reservation
3. Display Available Flights
4. Display Reservations
0. Exit

Enter your choice: 0
Exiting the program...

--------------------------------
Process exited after 42.07 seconds with return value 0
Press any key to continue . . .
```